IN THE CLAIMS

Please amend the claims as follows:

1. - 13. (Cancelled)

14. (Currently Amended) A method, comprising:

storing one or more prior pop pointer values of a pop pointer;

processing one or more pop requests to read data from a FIFO memory;

receiving information to indicate at least one of the one or more pop requests was speculative and to indicate that a state of the pop pointer of the FIFO memory should be restored; and

selectively controlling an order in which a memory will be accessed by restoring one of the one or more prior pop pointer values to the pop pointer in response to the information.

15. (Original) The method of claim 14, wherein

the one or more prior pop pointer values of the pop pointer are stored into a pointer memory.

16. (Previously Presented) The method of claim 15, wherein

the restoring of the one prior pop pointer value to the pop pointer includes;

reading the one prior pop pointer value from the pointer memory, and

loading the one prior pop pointer value into the pop pointer.

17. (Previously Presented) The method of claim 14, comprising:

prior to the processing of the one or more pop requests,

storing data into a memory array of the FIFO memory, and

incrementing a push pointer.

18. (Previously Presented) The method of claim 17, comprising:

reading a pop pointer value of the pop pointer and a push pointer value of the push pointer, and

determining a status of the memory array in response to the pop pointer value and the push pointer value.

19. (Previously Presented) The method of claim 18, where,

the determining of the status of the memory array is further in response to a high threshold level and a low threshold level.

20. (Currently Amended) ~~The method of claim 19, where,~~

A method, comprising:

storing one or more prior pop pointer values of a pop pointer;

prior to processing one or more pop requests,

storing data into a memory array of a FIFO memory, and

incrementing a push pointer;

processing one or more pop requests to read data from the FIFO memory;

reading a pop pointer value of the pop pointer and a push pointer value of the push pointer;

determining a status of the memory array in response to the pop pointer value, the push pointer value, a high threshold level, and a low threshold level, where the high threshold level is responsive to the lesser of a maximum branch resolution latency and the low threshold level;

receiving information to indicate at least one of the one or more pop requests was speculative and to indicate that a state of the pop pointer of the FIFO memory should be restored; and

restoring one of the one or more prior pop pointer values to the pop pointer in response to the information.

21. (Previously Presented) The method of claim 20, where,

the maximum branch resolution latency is a depth of an instruction pipeline in a processor, the processor to couple to the FIFO memory.

22. (Previously Presented) The method of claim 20, where,

the information includes a branch resolution latency, the branch latency being the number of instruction cycles to resolve a conditional branch instruction in a processor, the processor to couple to the FIFO memory.

23. - 32. (Cancelled)

33. (Currently Amended) A processing unit, comprising:

a plurality of processors, each of the processors including an instruction pipeline to speculatively execute instructions before a conditional branch is resolved;

a first plurality of branch-aware first-in first-out (FIFO) memories to pass data from one processor to the next in a first direction, each branch-aware FIFO memory of the first plurality of branch aware FIFO memories interleaved between a pair of processors of the plurality of processors;

a first input branch-aware FIFO memory coupled to a first processor of the plurality of processors to receive input data in the processing unit;

a first output FIFO memory coupled to a last processor of the plurality of processors to output data from the processing unit;

a second plurality of branch-aware FIFO memories to pass data from one processor to the next in a second direction, each branch-aware FIFO memory of the second plurality of branch-aware FIFO memories interleaved between a pair of processors of the plurality of processors;

a second input branch-aware FIFO memory coupled to a last processor of the plurality of processors to receive input data in the processing unit; and

a second output FIFO memory coupled to the first processor of the plurality of processors to drive output data from the processing unit, where the output controls, at least in part, a memory access process.

34. (Cancelled)

35. (Previously Presented) The processing unit of claim 33, where,

each branch-aware FIFO memory includes,

a memory array to store data;

a push pointer coupled to the memory array to address memory locations therein to write data;

a pop pointer coupled to the memory array to address memory locations therein to read data;

a pointer memory coupled to the pop pointer, the pointer memory to save one or more prior pop pointer values of the pop pointer; and

control logic coupled to the pointer memory, the control logic to restore one of the one or more prior pop pointer values to the pop pointer in response to branch information received from a processor.

36. (Cancelled)

37. (Currently Amended) A computer system, comprising:

an input/output device;

a dynamic random access memory; and

a multi-processor coupled to the dynamic random access memory and the input/output device, the multi-processor including,

a plurality of processors, each of the processors including an instruction pipeline to speculatively execute instructions before a conditional branch is resolved;

a first plurality of branch-aware first-in first-out (FIFO) memories to pass data from one processor to the next in a first direction, each branch-aware FIFO memory of the first plurality of branch-aware FIFO memories interleaved between a pair of processors of the plurality of processors;

a first input branch-aware FIFO memory coupled to a first processor of the plurality of processors to receive input data in the multi-processor ~~processing unit~~;

a first output FIFO memory coupled to a last processor of the plurality of processors to drive output data from the multi-processor ~~processing unit,~~ where the output controls, at least in part, access to the dynamic random access memory, the output depending, at least in part, on a status of the first output FIFO memory as determined with respect to a high threshold level and a low threshold level, the high threshold level being responsive to the lesser of a maximum branch resolution latency and the low threshold level; and

wherein each branch-aware FIFO memory includes,

a memory array to store data,

a push pointer coupled to the memory array to address memory locations therein to write data,

a pop pointer coupled to the memory array to address memory locations therein to read data,

a pointer memory coupled to the pop pointer, the pointer memory to save one or more prior pop pointer values of the pop pointer, and

control logic coupled to the pointer memory, the control logic to restore one of the one or more prior pop pointer values to the pop pointer in response to branch information received from a processor,

wherein the multi-processor further includes,

a second plurality of branch-aware FIFO memories to pass data from one processor to the next in a second direction, each branch aware FIFO memory of the second plurality of branch-aware FIFO memories interleaved between a pair of processors of the plurality of processors;

a second input branch-aware FIFO memory coupled to a last processor of the plurality of processors to receive input data in the multi-processor ~~processing unit~~; and

a second output FIFO memory coupled to the first processor of the plurality of processors to drive output data from the multi-processor ~~processing unit~~.

38. - 47. (Cancelled)

48. (Currently Amended) A processor comprising:

an instruction pipeline to speculatively execute instructions before a conditional branch is resolved;

a first branch-aware first-in first-out (FIFO) memory to pass data from the processor to another processor, the first branch-aware FIFO memory to receive branch information responsive to the conditional branch, where the data to be passed from the processor to another processor controls, at least in part, a branch operation, the data depending, at least in part, on a status of the first FIFO memory as determined with respect to a high threshold level and a low threshold level, the high threshold level being responsive to the lesser of a maximum branch resolution latency and the low threshold level;

the first branch-aware FIFO memory including:

a memory array to store data,

a push pointer coupled to the memory array to address memory locations therein to write data,

a pop pointer coupled to the memory array to address memory locations therein to read data,

a pointer memory coupled to the pop pointer, the pointer memory to save one or more prior pop pointer values of the pop pointer, and

control logic coupled to the pointer memory, the control logic to restore one of the one or more prior pop pointer values to the pop pointer in response to the branch information; and

a second branch-aware FIFO memory to pass data from another processor to the processor.

49. (Previously Presented) The processor of claim 48, where,

the second branch-aware FIFO memory is to receive branch information responsive to a second conditional branch, and

the second branch-aware FIFO memory includes:

a second memory array to store data,

a second push pointer coupled to the second memory array to address memory locations therein to write data,

a second pop pointer coupled to the second memory array to address memory locations therein to read data,

a second pointer memory, coupled to the second pop pointer, the second pointer memory to save one or more prior second pop pointer values of the second pop pointer, and

a second control logic coupled to the second pointer memory, the second control logic to restore one of the one or more prior second pop pointer values to the second pop pointer in response to the second branch information.

50. (Currently Amended) An apparatus, comprising:

a plurality of storage locations to store data;

a pointer storage area to store one or more pointer values to index the plurality of storage locations; and

a control logic:

to control storing a pointer value to the pointer storage area based, at least in part, on program branching information, and

to control retrieving a pointer value from the pointer storage area based, at least in part, on program branching information,

where the control depends, at least in part, on a status of the pointer storage area as determined with respect to a high threshold level and a low threshold level, the high threshold level being responsive to the lesser of a maximum branch resolution latency and the low threshold level.

51. (Previously Presented) The apparatus of claim 50, where the one or more pointer values include a pop pointer to index data to be read from the plurality of storage locations and a push pointer to index data to be stored to the plurality of storage locations.

52. (Previously Presented) The apparatus of claim 50, where the program branching information includes a branch flag to indicate a condition in which the one or more pointer values are to be read from the pointer storage area.

53. (Previously Presented) The apparatus of claim 50, where the program branching information corresponds to information speculatively read from one or more of the plurality of storage locations.

54. (Previously Presented) The apparatus of claim 51, comprising:

a status logic to indicate an amount of information stored in the pointer storage area.

55. (Previously Presented) The apparatus of claim 54, the status logic to set a high status flag in response to the amount of information stored in the pointer storage area being greater than or equal to a high threshold level, and less than or equal to a maximum utilization level.

56. (Previously Presented) The apparatus of claim 55, the status logic to set a low status flag in response to an amount of information stored in the pointer storage area being less than or equal to a low threshold level, and greater than or equal to an empty threshold level.

57. (Previously Presented) The apparatus of claim 50, where the program branching information includes a branch resolution latency corresponding to a number of processor cycles to be used in resolving a conditional branch instruction.

58. (Currently Amended) A processor comprising:

a plurality of processor cores, each of the processor cores including an instruction pipeline to speculatively execute instructions before a conditional branch is resolved;

a first plurality of first-in first-out (FIFO) memories to store data to be passed between the plurality of processor cores, wherein the first plurality of FIFO memories includes:

a first input FIFO memory to store input data to be processed by at least one of the plurality of processor cores, and

a first output FIFO memory to store output data to be output from at least one of the plurality of processor cores,

where the output data controls, at least in part, access to a memory, the output depending, at least in part, on a status of the first input FIFO memory as determined with respect to a high threshold level and a low threshold level, the high threshold level being responsive to the lesser of a maximum branch resolution latency and the low threshold level and on a status of the first output FIFO memory as determined with respect to the high threshold level and the low threshold level.

59. (Previously Presented) The processor of claim 58, where each FIFO memory includes:
a push pointer storage area to store an address to which data is to be written.

60. (Previously Presented) The processor of claim 59, where each FIFO memory includes:
a pop pointer storage area to store an address memory location from which data is to be read.

61. (Previously Presented) The processor of claim 60, where each FIFO memory includes:
a pointer memory to save one or more prior pop pointer values.

62. (Previously Presented) The processor of claim 61, where each FIFO memory includes:
control logic to retrieve one of the one or more prior pop pointer values in response to branch information.

63. (Previously Presented) The processor of claim 58, where each FIFO memory includes:
a push pointer storage area to store an address to which data is to be written.

64. (Previously Presented) The processor of claim 63, where each FIFO memory includes:
a pop pointer storage area to store an address from which data is to be read.

65. (Previously Presented) The processor claim 64, where each FIFO memory includes a pointer memory to store one or more prior pop pointer values.

66. (Previously Presented) The processor of claim 65, where each FIFO memory includes: control logic to retrieve one of the one or more prior pop pointer values in response to branch information.

67. (Previously Presented) A processor, comprising:

a memory to store instructions;

a plurality of processor cores, each of the processor cores including an instruction pipeline to speculatively execute the instructions before a conditional branch is resolved;

a first plurality of first-in first-out (FIFO) memories to store data to be passed between the plurality of processor cores, wherein the first plurality of FIFO memories includes a first input FIFO memory to store input data to be processed by at least one of the plurality of processor cores and a first output FIFO memory to store data to be output data from at least one of the plurality of processor cores.

68. (Previously Presented) The processor of claim 67, where each FIFO memory includes a push pointer storage area to store an address to which to which data is to be written.

69. (Previously Presented) The processor of claim 68, where each FIFO memory includes a pop pointer storage area to store an address memory location from which data is to be read.

70. (Previously Presented) The processor of claim 69, where each FIFO memory includes a pointer memory to save one or more prior pop pointer values.

71. (Previously Presented) The processor of claim 70, where each FIFO memory includes control logic to retrieve one of the one or more prior pop pointer values in response to branch information.

72. (Previously Presented) The processor of claim 67, where each FIFO memory includes a memory array to store data.

73. (Previously Presented) The processor of claim 67, where each FIFO memory includes a push pointer storage area to store an address to which data is to be written.

74. (Previously Presented) The processor of claim 73, where each FIFO memory includes a pop pointer storage area to store an address from which data is to be read.

75. (Previously Presented) The processor of claim 74, where each FIFO memory includes a pointer memory to store one or more prior pop pointer values.

76. (Previously Presented) The processor of claim 75, where each FIFO memory includes control logic to retrieve one of the one or more prior pop pointer values in response to branch information.